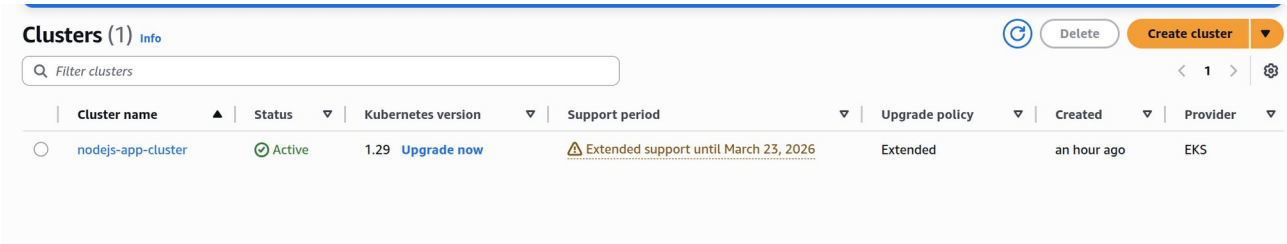
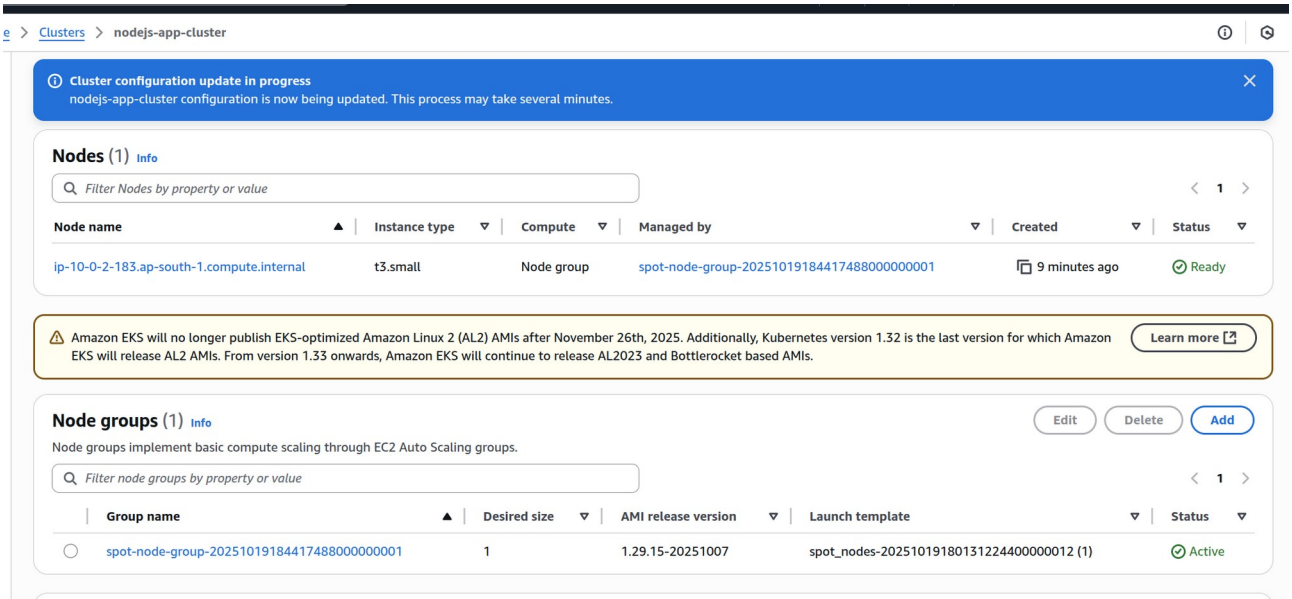
1. Cluster -



2. Spot instance -



3. Spot instance on cli as k8s node -

```
$ sudo kubectl get nodes
NAME                                          STATUS   ROLES    AGE   VERSION
ip-10-0-2-183.ap-south-1.compute.internal   Ready    <none>   18m   v1.29.15-eks-113cf36
$ _
```

4. Deploy mongodb -

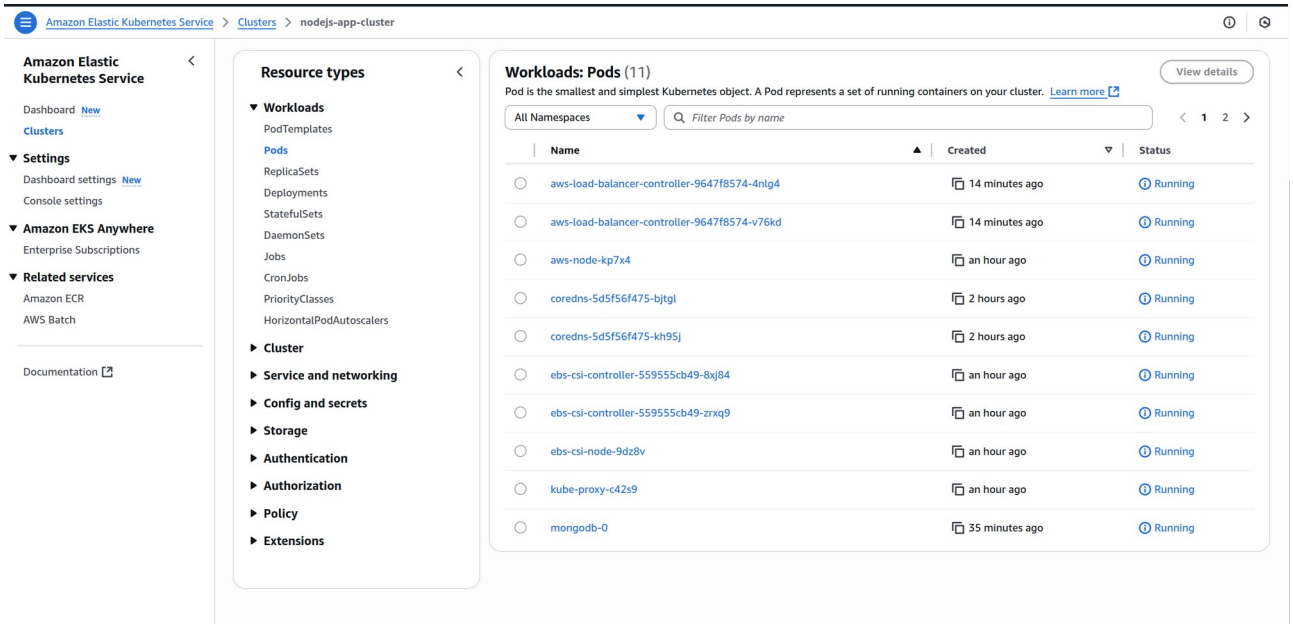
```
$ sudo kubectl apply -f mongo-secrets.yaml,mongodb-deployment.yaml
secret/mongodb-secret created
storageclass.storage.k8s.io/ebs-sc created
service/mongodb-service created
statefulset.apps/mongodb created
$ sudo kubectl get pvc -l app=mongodb
NAME                         STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
mongodb-storage-mongodb-0    Bound    pvc-3840c369-7a0d-43e4-a6cb-f9da3041d57c   1Gi        RWO            ebs-sc         <unset>                 12s
$ sudo kubectl get storageclass
NAME            PROVISIONER             RECLAIMPOLICY   VOLUMEBINDINGMODE      ALLOWVOLUMEEXPANSION   AGE
ebs-sc          ebs.csi.aws.com         Delete          WaitForFirstConsumer   false                  22s
gp2 (default)   kubernetes.io/aws-ebs   Delete          WaitForFirstConsumer   false                  71m
$ sudo kubectl get statefulsets
NAME      READY   AGE
mongodb   1/1     30s
$ _
```

5. Deploy nodejs -

```
$ sudo kubectl apply -f nodejs-deployment.yaml
service/nodejs-service created
deployment.apps/nodejs-app-deployment created
$ sudo kubectl get pods
NAME                                  READY   STATUS              RESTARTS   AGE
mongodb-0                             1/1     Running             0          80s
nodejs-app-deployment-56c95c79b4-7xxx9   0/1     ContainerCreating   0          6s
$ sudo kubectl get pods
NAME                                  READY   STATUS    RESTARTS   AGE
mongodb-0                             1/1     Running   0          102s
nodejs-app-deployment-56c95c79b4-7xxx9   1/1     Running   0          28s
$ _
```

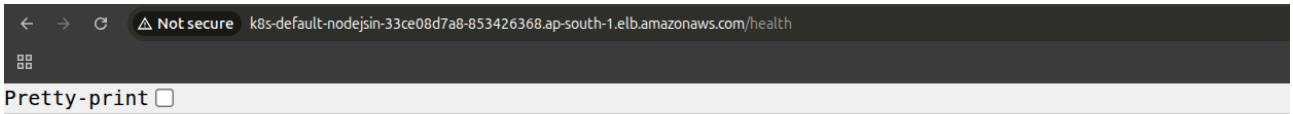
6. Deploy alb -

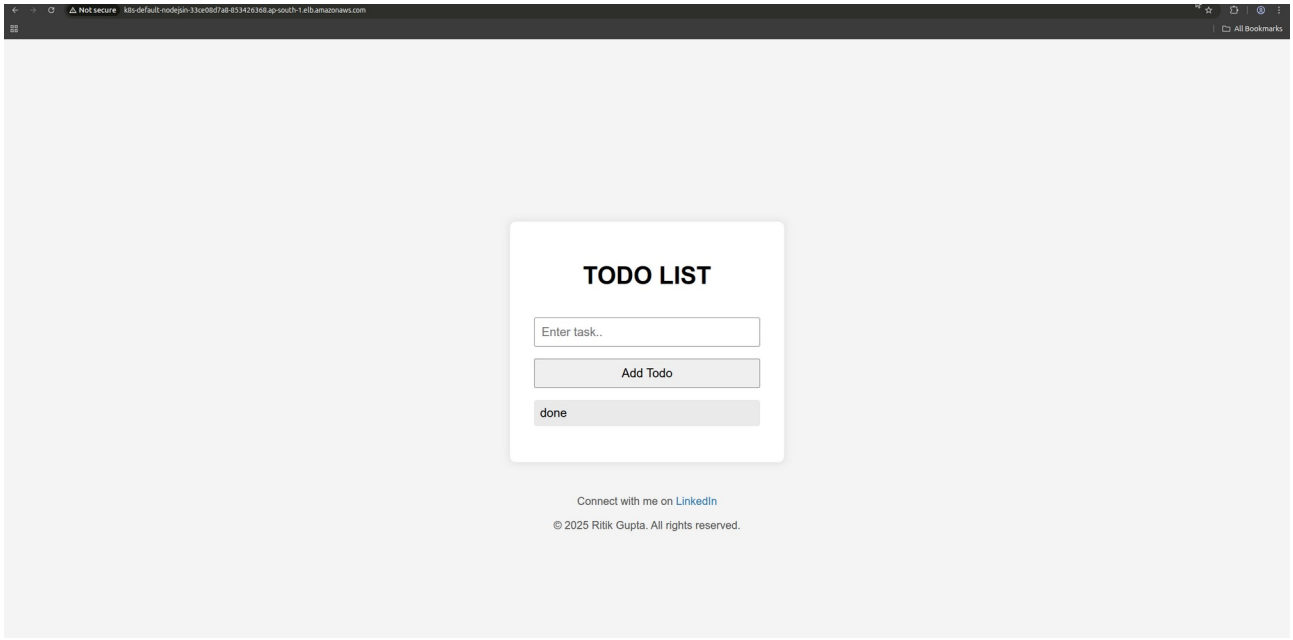
```
$ sudo kubectl apply -f ingress.yaml
Warning: annotation "kubernetes.io/ingress.class" is deprecated, please use 'spec.ingressClassName' instead
ingress.networking.k8s.io/nodejs-ingress created
$ _
```



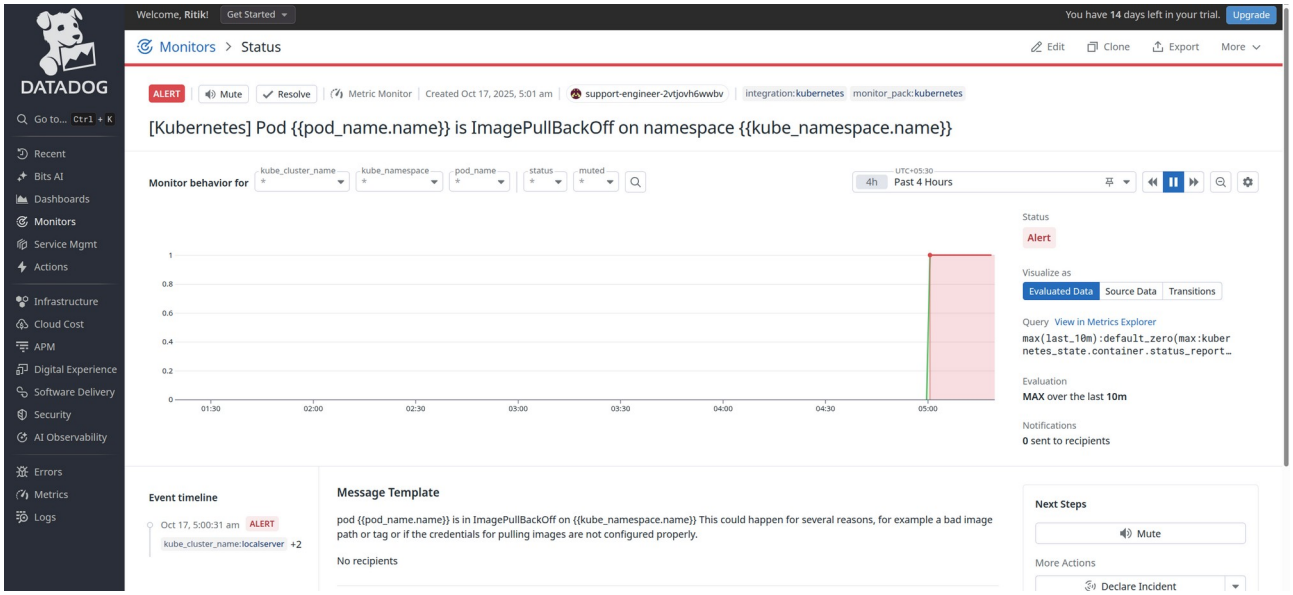

```
$ sudo kubectl describe ingress nodejs-ingress
Name:             nodejs-ingress
Labels:           <none>
Namespace:        default
Address:          k8s-default-nodejsin-33ce08d7a8-853426368.ap-south-1.elb.amazonaws.com
Ingress Class:    <none>
Default backend:  <default>
Rules:
  Host        Path  Backends
  ----        ----  --------
  *
              /   nodejs-service:80 (10.0.2.91:3000)
Annotations:  alb.ingress.kubernetes.io/scheme: internet-facing
              alb.ingress.kubernetes.io/target-type: ip
              kubernetes.io/ingress.class: alb
Events:
  Type    Reason                  Age    From     Message
  ----    ------                  ----   ----     -------
  Normal  SuccessfullyReconciled  8m16s  ingress  Successfully reconciled
$ sudo kubectl get ingress nodejs-ingress
NAME           CLASS    HOSTS   ADDRESS                                                         PORTS   AGE
nodejs-ingress   <none>   *       k8s-default-nodejsin-33ce08d7a8-853426368.ap-south-1.elb.amazonaws.com   80      8m25s
$
```

8. Access the endpoint -





9. Datadog alert -



10. Delete resources -